COLOR CORRECTION FOR IMAGES FORMING A PANORAMIC IMAGE

Junxing Shen

Yushan Huang

FIELD OF INVENTION

[0001] This invention relates to color correction to images to make up a panoramic image.

DESCRIPTION OF RELATED ART

[0002] When a photographer captures a scene using a camera, the desired field of view may be larger than the normal field of view of the camera. Digital photography allows a panoramic image to be produced without the need of purchasing special equipment such as a panoramic camera or fisheye lenses. For example, a photographer with a digital camera may capture a series of digital pictures of a scene by rotating the camera and taking pictures in a sequence. The captured images may then be stitched together to produce a panoramic picture of the scene. Similarly, film-based photographs can be digitized, and the panoramic picture can be composed by stitching together the digitized images. Presently, digital image programs are available for stitching multiple digital images together to form a panoramic picture. Exemplary programs include Ulead Cool 360®, Live Picture PhotoVista®, and MGI PhotoSuite III®.

[0003] Even when the stitching process works successfully, one may notice that the colors in the images do not match. This is due to differences in exposure and in the white balancing between the images. Thus, what is needed is a method to correct the colors in the images that make up the panoramic image to make the panoramic more realistic.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The patent or application file contains at least one drawing executed in color. Copies of this patent or patent application publication with color drawings will be provided by the Office upon request and payment of the necessary fee.

[0005] Fig. 1 illustrates two images stitched together to form a panoramic image in one embodiment of the invention.

[0006] Fig. 2 is a flowchart of a method for color matching the two images of Fig. 1 in one embodiment of the invention.

[0007] Fig. 3 illustrates outliers discarded by the method of Fig. 2 in one embodiment of the invention.

[0008] Fig. 4 illustrates the concept of histogram matching in the method of Fig. 2 in one embodiment of the invention.

[0009] Figs. 5 and 6 illustrate the implementation of histogram matching in the method of Fig. 2 in one embodiment of the invention.

[0010] Fig. 7 is a flowchart of a method for optimizing a golden section search of the method of Fig. 3 in one embodiment of the invention.

[0011] Figs. 8, 9, and 10 illustrate the result of color matching two images stitched together to form a panoramic image using the method of Fig. 2 in one embodiment of the invention.

SUMMARY

[0012] In one embodiment of the invention, a method for color matching a first image and a second image, wherein a first region of the first image and a second region of the second image overlap, includes removing overlapping pixels in the first and the second regions that have pixel values are too different, generating a first histogram of the first region, generating a second histogram of the second region, determining corresponding pixel values in the first and the second histograms, determining parameters of an optoelectronic conversion function that matches the corresponding pixel values, and color matching the second image to the first image by applying the optoelectronic conversion function with the determined parameters to the second image.

DETAILED DESCRIPTION

[0013] Fig. 1 illustrates a panoramic image 30 consisting two images 32 and 34 that are stitched together in one embodiment of the invention. A panoramic program has been used to determine

a region 36 of image 32 and a region 38 of image 34 that overlap. The same panoramic program has stitched (i.e., aligned) images 32 and 32 to overlap in regions 36 and 38.

[0014] Fig. 2 is a flowchart of a method 50 for a panoramic program to color match images (e.g., images 32 and 34 in Fig. 1) that overlap in one embodiment of the invention. The panoramic program can be executed by any computer system (e.g., a processor and a memory).

[0015] In step 52, the system discards pairs of overlapping pixels in regions 36 and 38 that are unreliable for color matching. A pair of overlapping pixels is unreliable if the difference of brightness, (e.g., $(R + G + B) / 3$), between the overlapping pixels is too great. In one embodiment, a pair of overlapping pixels is unreliable when it is among the 10 percent of the overlapping pixels that have the highest brightness difference. Fig. 3 illustrates an example of discarded overlapping pixels in regions 36 and 38. It can be seen that most discarded overlapping pixels are part of moving objects, edges, and regions with abundant texture that cannot be aligned between regions 36 and 38.

[0016] In step 54, the system performs a conventional histogram matching for the remaining overlapping pixels in regions 36 and 38. The concept of histogram matching is first explained in reference to Fig. 4. First, the system generates a histogram 72 for region 36 and a histogram 74 for region 38 as shown. Each histogram has pixel value intensity bins that store the units of pixels that have the corresponding pixel values of the bins. The bins of histogram 72 and laid end-to-end in ascending (or descending) bin order to form a series 76. Similarly the bins of histogram 74 are laid end-to-end in ascending bin order to form a series 78. The corresponding pixels between regions 36 and 38 are then assumed to have the pixel values in the corresponding horizontal locations along series 76 and 78 regardless of their actual positions in regions 36 and 38.

[0017] One implementation of histogram matching is now explained in reference to Figs. 5 and 6. In method 80, parameters i and j indicate the bins in histogram 72A and 74A, parameters *rem1* and *rem2* indicate the remaining numbers of pixels in the bins, and *min_rem* indicates the minimum remaining number between *rem1* and *rem2*.

[0018] In step 81, the system initializes all the entries in a lookup table (LUT) 79. Each entry LUT[i][j] in LUT 79 records the number of pairs of corresponding pixel values having a pixel value corresponding to a bin i in histogram 72A and a pixel value corresponding to a bin j in histogram 74A.

[0019] In step 82, the system initializes the first loop through method 80. Specifically, the system sets $i = 0$ and $j = 0$. The system also sets a parameter $rem1 = $ h1[0] and rem2 = h2[0], where h1[] is the number of pixels having a certain pixel value in histogram 72A, and h2[] is the number of pixels having a certain pixel value in histogram 74A.

[0020] In step 83, the system updates the correspondence between the pixels in the current bins of histogram 72A and 74A in LUT 79 by setting $min_rem = $ min($rem1$, $rem2$) and incrementing LUT[i][j] by min_rem. The system also decrements the remaining numbers of pixels in the current bins by setting $rem1 = rem1 - min\_rem$, and $rem2 = rem2 - min\_rem$.

[0021] In step 84, the system determines if the current bin in histogram 72A is empty because its pixels have all been matched to corresponding pixels in histogram 74A. The system does this by determining if $rem1 = 0$. If so, step 84 is followed by step 85. Otherwise step 84 is followed by step 86.

[0022] In step 85, the system selects the next bin in histogram 72A by incrementing parameter i. The system also tracks the number of pixels remaining in the next bin by setting $rem1 = $ h1[i]. Step 85 is followed by step 88.

[0023] In step 86, the system determines if the current bin in histogram 74A is empty because its pixels have all been matched to corresponding pixels in histogram 72A. The system does this by determining if $rem2 = 0$. If so, step 86 is followed by step 87. Otherwise step 86 is followed by step 88.

[0024] In step 87, the system selects the next bin in histogram 74A by incrementing parameter j. The system also tracks the number of pixels remaining in the next bin by setting $rem2 = $ h2[j]. Step 87 is followed by step 88.

[0025] In step 88, the system determines if all the pixels in histograms 72A and 74A have been matched. The system does this by determining if $i < 256$ and $j < 256$. If so, then step 88 is followed by step 89, which ends method 80. Otherwise step 88 is followed by step 83 and method 80 repeats until the correspondence between the pixels in histograms 72A and 74A have been recorded in LUT 79.

[0026] For simplicity, only one histogram per overlapping region is shown. In actual application to color images, there would be a histogram for each color component per overlapping region. In other words, there would be a histogram for the R values, a histogram for the G values, and a histogram for the B values in region 36. Similarly, there would be a histogram for the R values, a histogram for the G values, and a histogram for the B values in region 38.

[0027] Referring back to step 56 of method 50 in Fig. 2, the system finds optimal parameters for an optoelectronic conversion function (OECF) that best matches the colors of regions 36 and 38. In one embodiment, the system uses a conventional Laguerre OECF is defined as follows:

$$S(x) = x + \frac{2}{\pi} \arctan\left(\frac{a \sin(\pi x)}{1 - a \cos(\pi x)} \right), \qquad (1)$$

where $S()$ is the actual pixel value of a pixel in a scene, x is the pixel value of that pixel in an image of the scene, and $a \in (-1,1)$ is an unknown parameter of the OECF. It has been mathematically proven that when parameter x is normalized to $(0,1)$, $S()$ is monotonic and its inverse is simply found by changing the sign of parameter a. To color match regions 36 and 38, the system minimizes a conventional color matching error defined as follows:

$$e = \sum_{x_1 \in \Re_1, x_2 \in \Re_2} \|S(x_1) - \tau S(x_2)\|^2, \qquad (2)$$

where e is the color matching error, R_1 and R_2 are the overlapping regions in two images to be stitched, x_1 and x_2 are the corresponding pixel values of a single pixel in the scene, $S()$ is the OECF, and τ accounts for the difference in exposure and white balancing of the two overlapping regions. If τ and the OECF are both unknown, then the minimum of color matching error e has a

trivial and degenerate solution: $\tau = 1, S(x) = const$. To avoid this problem, Equation 2 can be expressed in the equalized images space as follows:

$$e = \sum_{x_1 \in \Re_1, x_2 \in \Re_2} \left\| x_1 - S^{-1}(\tau S(x_2)) \right\|^2 , \qquad (3.1)$$

where S^{-1} is the inverse OECF. To be used with the LUT generated by histogram matching above, equation 3.1 is rewritten as:

$$e = \sum_{i=0}^{255} \sum_{j=0}^{255} LUT[i][j]((i+1)/256.0 - S^{-1}(\tau S((j+1)/256.0))) , \qquad (3.2)$$

In equation 3.2, LUT[i][j] is used to determine how many pairs of corresponding pixels that have the pixel value of bin i and the pixel value of bin j, and $(i + 1) / 256$ and $(j + 1) / 256$ are used to normalize the pixel values to determine their difference using the Laguerre OECF.

[0028] From experience, it has been determined that τ has a uni-mode effect on the color correction results, in that there is a single optimal τ that gives better results than all other values. Thus, the system performs a golden section search on Equation 3.2 to find optimal τ with high convergence speed.

[0029] Fig. 7 is a flowchart of a method 90 for optimizing the golden section search in one embodiment.

[0030] In step 92, the system initializes parameter a to -0.9, and a parameter *MinError* to a large number (e.g., 1.0e20). System uses parameter *MinError* to track the smallest color matching error e through iterations of method 90.

[0031] In step 94, the system performs a conventional golden section search with parameter a being fixed and parameter τ being varied.

[0032] In step 96, the system saves the best result of the golden section search. Specifically, if a color matching error e generated with a particular combination of parameters a and τ is less than parameter *MinError*, the system sets parameter *MinError* equal to that color matching error e

and saves the current parameters a as a parameter *BestA* and τ as a parameter *BestT*. This is repeated until the system has tried a range of parameter τ for the current parameter a.

[0033] In step 98, the system increments parameter a by 0.1.

[0034] In step 100, the system determines if parameter a is less than or equal to 0.9. If so, step 100 is followed by step 94 and method 90 repeats until the system has performed the gold section search for a range of parameters a and τ. If parameter a is not less than or equal to 0.9, step 100 is followed by step 102, which ends method 90 because the system has found the optimal parameters a and τ.

[0035] For simplicity, only one OECF and one color matching error e are listed above. In actual application to color images, there would be three OECFs and three color matching errors e for each color component. In other words, there would be S_R, a_R, τ_R, and e_R for the R values, S_G, a_G, τ_G, and e_G for the G values, and S_B, a_B, τ_B, and e_B for the B values.

[0036] Referring back to step 58 of method 50 in Fig. 2, the system then applies the OECF with the optimal parameters a and τ to color match image 34 to image 32. Specifically, the system applies the following color correction to image 34:

$$x_c = S^{-1}(\tau S(x_o)), \tag{4.1}$$

wherein x_c is the corrected pixel value of a pixel in image 34, and x_o is the original pixel value of a pixel in image 34. Equation 4.1 can be written for each color component as follows:

$$[R_c, G_c, B_c] = [S_R^{-1}(\tau_R S_R(R)), S_G^{-1}(\tau_G S_G(G)), S_B^{-1}(\tau_B S_B(B))], \tag{4.2}$$

wherein R_c, G_c, and B_c are the corrected pixel values of the pixels in image 34, and R, G, B are the original pixel values of the pixels in image 34.

[0037] A pixel saturation problem exists when applying Equation 4.2 for color matching image 34 to image 32. If parameter τ is greater than 1, $\tau S(x)$ will be easily greater than 1. Thus, τ has to be weighted according to the pixel value. In one embodiment, a linear weighing function is determined experimentally to give good result is defined as follows:

$$x_c = S^{-1}(W(\tau, x_o) S(x_o)), \text{ or} \tag{5.1}$$

$$[R_c, G_c, B_c] = [S_R^{-1}(W(\tau_R, R) S_R(R)), S_G^{-1}(W(\tau_G, G) S_G(G)), S_B^{-1}(W(\tau_B, B) S_B(B))], \tag{5.2}$$

where W is the weighing function defined as:

$$W(\tau, x) = \tau + (1 - \tau)x. \tag{6}$$

[0038] Figs. 8 and 9 illustrate images 112 and 114 that are to be stitched together and then color matched using method 50 described above to form a panoramic image 116 in Fig. 10 in one embodiment of the invention. As can be seen, color consistency between images 112 and 114 has been greatly improved with method 50.

[0039] Various other adaptations and combinations of features of the embodiments disclosed are within the scope of the invention. Numerous embodiments are encompassed by the following claims.